# AMENDMENTS TO THE SPECIFICATION

Please amend paragraph [0006] as follows:

[0006] In an embodiment of the invention, a cache list is created comprising a list of all test components which will be delivered to the testing stations and which are necessary to display the items to the user. Test components are used to create test items to be delivered to the user. The cache list is delivered to the testing station and synchronized with a central server. The testing station then pre-fetches a configurable number of components <u>from</u> ~~on~~ the cache list. Upon the occurrence of certain pre-determined trigger events, additional <u>cache</u> components are delivered from the central server to the testing stations in order to <u>eliminate</u> ~~minimize~~ data latency issues, <u>which arise when components are fetched only at the instant they are needed for display</u>.

Please amend paragraph [0027] as follows:

[0027] The web server 122 is in communication with the testing stations 111a. . .n to provide access to the service center 120. The web server 122 is in further communication with a data storage device 123. The data storage device 123 is configured to store all data received from the testing stations 111a...n which information may comprise the user's responses to questions and any other interaction between the user and the testing station 111a...n. In addition, the data storage device 123 stores data necessary for fault recovery purposes in the <u>event</u> ~~even~~ the testing station 111a...n fails <u>or the network connection is broken</u>. Fault recovery information may include a substantial mirror copy of the current state of a test being administered to user sufficient to allow fault recovery without reconstructing the state from stored information. State information may include test answers and elapsed testing time for a test currently being administered to a user, for example, on a testing station. The data storage device 123 also caches test information (e.g., test questions, etc.) received from the backend 130 and is further configured to store test items which may be delivered to and displayed on the testing stations 111a...n and information received from the testing stations 111a...n (e.g., test results, user identifying information, etc.) to be transmitted to the backend 130.

Please amend paragraph [0028] as follows:

-2-

[0028] The application server 125 is configured to support the testing software 112 operating on the testing stations 111a...n. The application service 125 may receive requests from the testing stations 111a...n for data and appropriately respond to such requests. The application server 125 will control the delivery of test items and components to the testing stations 111a...n, and may provide services, for example ~~via~~ an ENTERPRISE JAVA BEANS (EJB) container ~~or alternately an Active X control~~. Furthermore, data transmitted from the testing stations 111a...n is received via the web server 122 and is delivered to the data storage device 123.

On Page 9, please amend the following paragraph:

Backend 130

The data storage device 123 receives data from and delivers data to the backend 130. The backend 130 is the location at which the data received from the testing stations such as the user's information and test results are stored and at which the test information is prepared and packaged. In addition, the backend 130 ~~package migration tool 132~~ stores the latest release of the test content (items) ~~packages~~ to be delivered to the data storage device 123. Test items are stored on a data storage backend device 133. In an embodiment of the invention, the test items are grouped together and stored on the data storage backend device 133 in the form of test packages. The test items are retrieved from the data storage backend device 133 and properly formatted for delivery to the data storage device 123 by the package migration tool 132. When test items are stored on the data storage backend device 133 in the form of test packages, the package migration tool 132 also manages the proper handling of subsequent versions of test packages to ensure that the proper version is being accessed from the data storage device 123. For example, test package A (Ver. 1) may be created on data storage device 133 and delivered by the package migration tool 132 to the data storage device 123. Subsequently test package A may then be updated as Ver. 2. When the package migration tool 132 delivers the Ver. 2 to the data storage device 123, it does so with an indicator that the prior version (i.e., Ver. 1) should be marked undeliverable and in response to requests for test package A, Ver. 2 should be provided. The package migration tool 132 may also perform encryption by any conventional encryption method, such as those based on symmetric key algorithms or public/private key algorithms.

Please amend paragraph [0031] as follows:

[0031] The test state data may comprises the elapsed test time, the current assessment item, prior assessment items, and responses to prior assessment items. The elapsed test time may includes an amount of test taking time that has elapsed from a starting time of a test being administered. The elapsed test time takes into consideration any amount of time that a test is unavailable to the user while the test is being administered. For example, portions of the test may be downloaded during administration of the test. Due to transmission latency, questions may not be available to the user. If the user is allotted a predetermined amount of time to take the test, the user will not be penalized for the time the questions are not available to the user. Also included in the test state are the user's responses to prior assessment items. This includes any answer the user has entered.

Please amend paragraph [0032] as follows:

[0032] FIG. 2 illustrates the method for performing state management according to an embodiment of the invention. The method is initiated at the start of the test 205 on the testing station. The testing station transmits identification of the examination being administered to the service center 210. The service center then creates an initial state object for that examination 220 and stores the initial state object on a data storage device. The initial state object is delivered from the service center to the testing station 225. The state object includes identification of the examination being administered any may also comprise test items, test item response, lapsed time and any other information related to the taking of the examination. A heartbeat timer is then initialized 230 and each user interaction with the testing station is recorded 235. Next, is the user interaction is checked to determine if it is one which has been be predetermined to cause the state to be updated 240. The user interactions which may cause the state to be updated can be configured to meet the needs of a given examination and may include any interaction or only those which result in a response being made to the current test item being displayed. If the user interaction is one requiring the state to be updated, the testing station then transmits the changes to the state to the service center 245. Importantly, only the changes to the state are transmitted, not the entire state object. The changes to the state which are transmitted to the service center include not only the user interaction causing the transmission at step 245 but

-4-

any other changes which may have occurred since the last transmission and the change in time. After the state object is updated with the changes in the state, the heartbeat timer is re-initialized 230

Please amend paragraph [0034] as follows:

[0034] FIG. 3 is a diagram illustrating a method for performing state management at the service center 120. The methods commences with the receipt of a test identification 305 from a testing station. The service center then creates an initial state objection 310 and delivers the initial state object to the testing station 315. Next, a fault timer is initialized 320 and the service center awaits data from the testing station. If data is received 325, the changes to the state are recorded in a data storage device and the changes to the state are recorded 350. Alternatively, if data is not received 325, the fault timer is checked 330 to determine if the fault timer exceeds a maximum value. If the fault time does not exceed a maximum value, the fault timer is incremented 335 and the service center continues to await data from the testing station. If the fault timer does exceed the maximum value, the service center designates the that testing station session as restartable 340 and the testing session is closed 345.

Please amend paragraph [0035] as follows:

[0035] The fault timer will exceed a maximum value only when it has not received data from the testing station for a time period sufficient to conclude that the testing station is no longer operable. When the user reconnects, on the same, or a different testing station is restarted, the service center will identify the examination being conducted on the testing station as the examination previously closed and designated restartable. Using the initial state and the changes to the initial state which have been recorded on the data storage device, the service center will recreate the examination on the testing station at the point when the testing station failed, within parameters configurable to meet the requirements of the examination.

Please amend paragraph [0036] as follows:

[0036] The system 100 is operable to administer any type of examination including, but not limited to, linear tests and adaptive tests. For linear tests, the questions for the test are

predetermined. However, instead of transmitting an entire test to a testing station, the service center 120 may transmit the test in portions during administration of the test using the method described herein. For example, one portion of the test may be transmitted to the testing station and stored in local volatile memory, such that a user can begin taking the test. As the test is being administered, additional portions of the test are continually transmitted to the testing station and stored in volatile memory. Events that may trigger transmission of a portion of the test can include the user answering a predetermined number of questions or elapse of a predetermined amount of time.

Please amend paragraph [0037] as follows:

[0037] The package migration tool 132 can retrieve a test from the data storage backend 133. The package migration tool 132 may store the test in the data storage backend device 123 ~~and the web server 122~~. Then, the application <u>server</u> ~~engine~~ 125 transmits portions of the test to the testing station 111a..n as needed<u>, via the servlet engine and the web server 122</u>.

Please amend paragraph [0038] as follows:

[0038] For the adaptive test, the next question presented to the user is not predetermined; rather, it is based on a user's <u>responses</u> ~~response~~ to <u>previous</u> ~~the then current~~ test <u>questions</u> ~~question~~. For adaptive tests, the service center 120 transmits a set of test items or questions to the testing station 111a..n administering the test. The test taking software 112, then selects one or more of the questions to display to the user based on the user's response to ~~a~~ previous <u>questions</u> ~~question~~. The selected questions(s) are then presented to the user during the test to be answered. This process is repeated until completion of the test. The service center 120 may transmit the set of questions while the user is answering a question. Therefore, the user may not have to wait to read the next question.

Please amend paragraph [0041] as follows:

[0041] The test taking software 112 and the application server 125 together manage the implementation of an adaptive test item selection algorithm such as ~~the lookahead~~ <u>a "lookahead"</u>

function described in Fig. 4. Any adaptive test algorithm may be implemented on the application server 125 and the test taking software 112. It selects test questions to be retrieved from the service center and to be displayed to the user based on the response to a previous <u>questions</u> <s>question</s>. The application server 125 may employ a mechanism for <u>"looking ahead" one or more items, in order to supply</u> <s>supplying</s> several questions to the testing stations 111a...n. For example, <u>prior to receiving an answer to the current item,</u> two <u>potential</u> sets of questions may be delivered to the testing station – one if the current item is answered correctly, and one if it is answered incorrectly. This is a lookahead algorithm for only one level of questions. The application may be configured to look further ahead and provide multiple levels of sets of questions if required to improve performance. Additionally, the application server may use an algorithm based on the answer choice selections such as the lookahead function described in Fig. 4.

Please amend paragraph [0042] as follows:

[0042] FIG. 5 is a diagram illustrating a method for caching the components of a test which method may be used for any examination model, including but not limited to linear tests or adaptive tests. Following the start of the test 505, the service center creates a cache list 510 of all the components which will be delivered to the testing stations in order to administer the test. In a preferred embodiment, the cache list will include any data, messages, and other information necessary to create and display the test items <u>and any other content</u> of the examination and record the user's responses. The cache list identifies the components which need to be delivered to the testing station, i.e., the test items. The cache list is then delivered to the testing station 515 and stored in volatile memory. The cache list on the test station and at the service center are then synchronized 520. The testing station then pre-fetches a configurable amount data 525. The amount of data pre-fetched is a cushion which the method seeks to maintain in order to avoid data latency issues. The amount of data stored as a cushion is configured according to the needs of the examination being administered and the network's bandwidth capabilities. All user interactions with the testing station are monitored and recorded 530. If a user action is a trigger action 535, the method manages/updates the cache list 540 to maintain the data cushion established in the pre-fetch step. Primarily, managing the cache list will require that additional <u>components</u> <s>data</s> should be fetched from the service center. However, in order to avoid